RECEIVED
CENTRAL FAX CENTER
APR 2 6 2007

Remarks

The present amendment responds to the final Official Action dated November 27, 2006 and is filed in conjunction with the filing of an RCE. A petition for a two month extension of time to respond and authorization to charge Deposit Account No. 50-1058 the large entity extension fee of $450 accompany this amendment. The Official Action rejected claims 1-5 and 9-12 under 35 U.S.C. § 102(b) based on Lowell et al. U.S. Patent No. 3,623,017 (Lowell). Claims 6-8, 13, and 14 were rejected under 35 U.S.C. § 103(a) based on Lowell. Claims 15-22 were rejected under 35 U.S.C. § 103(a) based on Lowell in view of Ishikawa U.S. Patent No. 5,787,303 (Ishikawa). These grounds of rejection are addressed below.

Claims 8, 20, and 21 have been canceled without prejudice. Claims 1, 2, 4, 6, 7, 12, 13, 15, 16, and 22 have been amended to be more clear and distinct and new claims 28-30 have been added. Claims 1-7, 9-19, 22, and 28-30 are presently pending.

The Art Rejections

As addressed in greater detail below, Lowell and Ishikawa do not support the Official Action's reading of them and the rejections based thereupon should be reconsidered and withdrawn. Further, the Applicants do not acquiesce in the analysis of Lowell and Ishikawa made by the Official Action and respectfully traverse the Official Action's analysis underlying its rejections.

Claims 1-5 and 9-12 were rejected under 35 U.S.C. § 102(b) based on Lowell. Lowell describes a processor that upon detection of an "extended sequence instruction" causes a higher

speed clock to be switched in to replace a low speed clock used for normal timing of an arithmetic section. Lowell executes at the lower speed clock for "normal" instructions and executes the "extended sequence instructions" with the high speed clock. Lowell, Abstract. Lowell defines "extended sequence instructions" as being "multiply, divide, square root, etc." which are typically longer execution type instructions as compared to AND, OR, NOT, simple adds, and the like.

The longer execution type of instructions typically require multiple execution stages or multiple cycles of execution before generating a result which, according to Lowell, "During the execution of this type of instruction by the arithmetic section of the computer, it is necessary to interrupt the normal timing of the computer until the extended sequence instruction has been completed." Lowell suggests that by switching to a high speed clock for "extended sequence instructions", the interruption of normal timing may be minimized. Lowell further describes circuitry to switch to a high speed clock as input to the command generator "so that the command enable signals for the arithmetic section are produced at a faster rate". To produce signals at a "faster rate" as described Lowell requires the use of multiple clock pulses to execute an extended sequence instruction. Lowell, col. 1, lines 5-12 and lines 47-55. Lowell is silent on the type of pipelining, if any, used in the arithmetic section and operation of the "extended sequence instructions". Lowell is also silent with regard to the latency of pipeline stages of the processor using his invention. This complete lack of information on latency of pipeline stages demonstrates that Lowell is not concerned with changing the latency of pipeline stages.

In contrast to Lowell, the present invention describes a processor pipeline which adapts the latency of stages of the processor pipeline, including an adaptable fetch stage, an adaptable decode

stage, and an adaptable execution stage, dependent on a classification of an instruction fetched for execution. As recited in the amended claim 1, the classification of an instruction is done in "an adaptable decode stage for classifying and decoding the instruction" and "generating an instruction class indication", where the adaptable decode stage is a stage of the processor pipeline. Also in claim 1, "an adaptable pipeline control unit" is "responsive to the instruction class indication for adapting the latency of the adaptable fetch stage, the adaptable decode stage, and an adaptable execution stage dependent on the instruction class". This adaptation of the pipeline stages can be seen in Fig. 9B, for various classification indications.

For example, a class one indication for instruction A(1) generated in decode stage A(1) 939 causes the fetch stage C(3) 944, the decode stage B(2) 945, and the execute stage A(1) 946 to operate with a latency of three MCLK cycles 947-949. For instruction B(2) with a class two indication generated in decode stage B(2) 945, the fetch stage D(1) 951, the decode stage C(3) 952, and the execute stage B(2) operate with a latency of four MCLK cycles 954-957. Further, for instruction C(3) with a class three indication generated in decode stage C(3) 952, the fetch stage E(1) 959, the decode stage D(1), and the execute stage C(3) operate with a latency of six MCLK cycles 962-967. See, page 25, line 22 – page 28, line 22 and Figs. 9B and 10 of the present specification. A single clock, MCLK 926, is used and the "latency of the adaptable fetch stage, the adaptable decode stage, and the adaptable execution stage" is adapted "dependent on the instruction class" as claimed in claim 1.

Lowell does not teach and does not make obvious classifying an instruction based on an instruction's execution latency to generate an instruction class indication. Rather, Lowell classifies

an instruction on whether it is a normal instruction or an instruction which takes an "extended sequence" to operate. This type of decision does not require knowledge of latency of a pipeline stage in order to classify instructions accordingly. As noted above, Lowell is silent on the latency of pipeline stages. Lowell simply shows a switch which selects a low speed clock or a high speed clock as input to a command generator that directly interfaces to the arithmetic section. Lowell's approach does not affect the timing of his instruction register which is considered part of a fetch stage or affect the timing of his instruction decoder which is considered part of a decode stage. Consequently, Lowell does not teach and does not make obvious "adapting the latency of the adaptable fetch stage, the adaptable decode stage, and the adaptable execution stage dependent on the instruction class indication" as presently claimed in claim 1.

The Official Action rejected claims 6-8, 13, and 14 under 35 U.S.C. § 103(a) based on Lowell. Claim 8 has been canceled without prejudice making the rejection of claim 8 moot. Since dependent claims 2-5, 6, 7, 9-12, 28, and 29 depend from and contain all the limitations of the amended claim 1, claims 2-5, 6, 7, 9-12, 28, and 29 distinguish from the references in the same manner as claim 1 and are in order for allowance.

Further, claim 13 recites, "an instruction class adaptable pipeline processor supporting at least two classes of instructions with a first class operable with a first latency for fetch, decode, and execute pipeline stages of the adaptable pipeline and a second class operable with a second latency for fetch, decode, and execute pipeline stages and where, the first latency is shorter than the second latency". Lowell does not teach and does not make obvious "a first plurality of instructions used in the program operable with the first latency specified in a format encoding of

the first plurality of instructions as class 1 instructions and with a second plurality of instructions used in the program operable with the second latency specified in a format encoding of the second plurality of instructions as class 2 instructions". Lowell also does not teach and does not make obvious "modifying the application program by changing, where appropriate, the format encodings of class 1 instructions to format encodings that specify the same functionality of the original class 1 instructions but provide class 2 instruction indications when decoded to minimize power use while still meeting performance requirements of the application program" as presently claimed in claim 13. In more specific detail, Lowell does not teach changing the "format encodings" of extended sequence instructions, which the Examiner asserts that the extended sequence instructions are class 1 instructions, to "format encodings" of normal instructions "that specify the same functionality" of the extended sequence instructions but provide normal instruction indications when decoded. Lowell is silent on format encodings of instructions and provides no motivation to change the format encodings of instructions. Claim 13 is not obvious and is not taught by Lowell.

Since dependent claim 14 depends from and contains all the limitations of the amended claim 13, claim 14 distinguishes from the reference in the same manner as claim 13 placing claims 13 and 14 in order for allowance.

Claims 15-22 were rejected under 35 U.S.C. § 103(a) based on Lowell in view of Ishikawa. Claims 20 and 21 have been canceled without prejudice making the rejection of these claims moot. As addressed in detail above, Lowell does not teach and does not make obvious classifying an instruction based on the instruction's execution latency. Lowell also does not teach and does not

make obvious "adapting the latency of the adaptable fetch stage, the adaptable decode stage, and the adaptable execution stage dependent on the instruction class indication".

Ishikawa describes a VLIW architecture for adjusting individual instruction execution dependent upon operand dependency interlocks. Ishikawa describes a four issue VLIW which may contain instructions having different execution times. Ishikawa checks for operand usage between instructions in the VLIW and based on operand dependencies allows an instruction to proceed with execution or be held up from executing until an operand dependency is resolved. Ishikawa, col. 3, lines 20-36. Ishikawa does not teach and does not make obvious a VLIW processor having "executable function instructions located in multiple instruction slot format VLIWs, the class one instructions having a first execution latency and the class two instructions having a longer second execution latency". Ishikawa does not teach and does not make obvious "an adaptable VLIW fetch stage" or "an adaptable decode stage for ... generating an instruction class indication for each of the plurality of executable function instructions". Ishikawa also does not teach and does not make obvious "an adaptable pipeline control unit" for "adapting the latency of the adaptable VLIW fetch stage, the adaptable decode stage, and the plurality of adaptable execution stages of the plurality of instruction class controllable pipelines dependent on the instruction class indications", as claimed in claim 15. Consequently, Ishikawa does not cure the deficiencies of Lowell.

Since dependent claims 16-19 and 22 depend from and contain all the limitations of the amended claim 15, respectively, claims 16-19 and 22 distinguish from the references in the same manner as claim 15 and claims 15-19 and 22 are in order for allowance.

Claim 30 has been newly added. Claim 30 includes the limitation of "an adaptable decode stage", "an adaptable execution stage", and "an adaptable pipeline control unit responsive to the instruction class indication" which are not taught and are not obvious from Lowell.

Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Respectfully submitted,

Peter H. Priest
Reg. No. 30,210
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713-7736
(919) 806-1600